Proceedings of the 2020 Winter Simulation Conference
K.-H. Bae, B. Feng, S. Kim, S. Lazarova-Molnar, Z. Zheng, T. Roeder, and R. Thiesing, eds.

FEASIBILITY DETERMINATION WHEN THE FEASIBLE REGION IS DEFINED BY NON-LINEAR INEQUALITIES

Daniel Solow

Weatherhead School of Management
Case Western Reserve University
10900 Euclid Avenue
Cleveland, OH 44106, USA

Roberto Szechtman

Department of Operations Research
Naval Postgraduate School
Glasgow Hall 254
Monterey, CA 93943, USA

Enver Yücesan

Technology and Operations Management Area
INSEAD
1, Ayer Rajah Avenue
Singapore 138676

ABSTRACT

A number of heuristic algorithms exist for determining whether each of r systems, characterized by m performance measures estimated through Monte Carlo simulation, belongs to a given set, which is defined by a finite collection of linear inequalities. The work here provides a heuristic for addressing a version of this problem in which the feasible region is defined by a finite collection of nonlinear inequalities. This approach allows the user to choose a desired level of confidence with which a system is correctly classified. The algorithm then uses appropriate-sized confidence rectangles centered at the estimated means to decide when a system can be classified with that level of confidence. While the worst-case behavior could potentially be bad, computational experiments show that the performance of the algorithm on randomly generated problems is satisfactory.

1 INTRODUCTION

We consider the problem of feasibility determination, i.e., the determination of whether any of the r systems with performance measures $\mu_1, \ldots, \mu_r \in R^m$, where $\mu_i = E[X_i]$ for some random vector $X_i \in R^m$, belong to a particular set. Given a set $\Gamma \subset R^m$ that is convex and closed with nonempty interior, system i is said to be "in Γ" if $\mu_i \in \Gamma$ and "not in Γ" if $\mu_i \notin \Gamma$. Unfortunately, the values of μ_i are unknown and must be estimated by Monte Carlo simulation through iid replicates of the random vectors X_i to form unbiased estimators $\bar{\mathbf{x}}^i(n_i) = n_i^{-1} \sum_{k=1}^{n_i} X_{i,k}$ of μ_i, for $i = 1, \ldots, r$. These estimates are then used to *classify* each system i as being in Γ or not in Γ. System i is said to be *correctly classified* if $\bar{X}_i(n_i)$ and μ_i are both in Γ or both not in Γ. Given a *computational budget* of N available replications for all of the r systems combined, the goal is to determine how "best" to use the budget, for example, to maximize the expected number of correctly classified systems.

Feasibility determination has many interesting applications such as the identification of a set of investment portfolios from a finite, but potentially large, set of investment opportunities whose expected payoff exceeds a desired threshold with a certain confidence. Another interesting application of the feasibility determination

problem is in agricultural research and development. The identification of robust seeds (e.g., against diseases or drought) necessitates the "crossing" of a large number of varieties until the desired traits can be obtained (Hunter and McClosky 2016). This combinatorial problem not only requires tens of millions of dollars, but also necessitates long development cycles of up to 12 years. A recent approach, *in-silico breeding,* relies on computer simulation modeling, enabling the assessment of the potential agronomic value of billions of genotypes even before any variety is physically planted in the field (Li et al. 2012). In other words, in-silico breeding allows for the identification of a feasible set of hybrid seeds (e.g., whose germination rate is superior to 95%) before expensive field trials are undertaken with varieties in that set.

An interesting problem in its own right, feasibility determination has recently attracted much attention within the context of ranking and selection (R&S) in the presence of stochastic constraints. To the best of our knowledge, Santner and Tamhane (1984) is the first paper to propose a two-stage procedure with a constraint on variance.Traditionally, three main approaches are used for the R&S problems with side constraints: the indifference-zone (IZ) approach, the optimal computing budget allocation (OCBA) approach, and the Bayesian approach. Within the IZ approach, Andradottir and Kim (2010) propose a two-phase algorithm whereby phase I identifies feasible systems while phase II selects the best among them in a setting where R&S is based on a primary performance measure subject to the feasibility of a (possibly correlated) secondary performance measure. Batur and Kim (2010) focus on feasibility determination and provide IZ procedures for identifying a set of feasible systems in the presence of multiple stochastic constraints. Within the OCBA approach, Lee et al. (2012) propose a budget allocation rule that maximizes the probability of correct selection of the best feasible system under multiple stochastic constraints. Similarly, Gao and Chen (2017) develop an OCBA procedure that returns a set of feasible systems in the presence of multiple stochastic constraints. Using Bayesian approach, Xie and Frazier (2013) present sampling procedures that compare multiple systems to a known standard on a single performance measure. This is essentially feasibility determination under a single stochastic constraint as the standard can be viewed as the constraint while a system that has a mean performance better than the standard can be classified as a feasible system. More recently, procedures that use the large deviation principle have been introduced by Szechtman and Yücesan (2016), Pasupathy et al. (2014), and Szechtman and Yücesan (2008) for finding a set of feasible systems. Gao and Chen (2017) also use the large deviation technique within the OCBA framework. Finally, He and Kim (2019) propose a Bayesian approach with a new reward function that favors the identification of "barely feasible/infeasible" systems whose mean performance is close to a threshold value of a given constraint.

For the setting where Γ is defined by a finite collection of linear inequalities, Solow et al. (2020) introduced efficient two-stage Bayesian and frequentist heuristics. In this paper, a new approach—in the form of a heuristic algorithm—is proposed for allocating the computational budget when the feasible region is defined by nonlinear inequalities. As with any heuristic, the proposed *Finite Budget Algorithm* (FBA) has its advantages and disadvantages. The primary advantage—and one of the main contributions—is that the FBA is capable of working with a set Γ that is defined by a finite collection of (convex) nonlinear, as well as linear, inequalities, which previously has not been possible. A disadvantage of the proposed algorithm is that, like the simplex algorithm for linear programming, in the worst-case, the behavior of FBA is bad—for example, by classifying a system incorrectly or by classifying a system with less confidence than that specified by the analyst. However, this worst-case behavior is mitigated with computational experiments on randomly generated problems to show that, like the simplex algorithm, the overall performance of the FBA is relatively good, both in terms of computational efficiency and the accuracy with which systems are classified.

The remainder of the paper is organized as follows. Section 2 introduces the notation and some preliminary results. We introduce our heuristic in Section 3. A numerical illustration is provided in Section 4. Section 5 closes the paper with concluding comments and future research directions..

2 NOTATION AND PRELIMINARY RESULTS

Throughout, all vectors are column vectors and the following notation is used consistently:

r	=	the number of independent systems.
m	=	the number of performance measures (and hence random variables) associated with each system.
s	=	the number of convex functions that define the feasible region
A^T	=	the transpose of the matrix A.
A_k	=	row k of the matrix A.
∂S	=	the boundary of a set $S \subset R^m$.
$\partial g(\mathbf{x})$	=	the set of subgradients of a convex function g at the point x.

2.1 Preliminary Results for Rectangles

Given vectors $\bar{\mathbf{x}}, \bar{\mathbf{d}} \in R^m$ with $\bar{\mathbf{d}} > \mathbf{0}$, for every real number $t \geq 0$, the m-dimensional rectangle of size t centered at $\bar{\mathbf{x}}$ in the direction $\bar{\mathbf{d}}$ is shown in Figure 1 for $m = 2$ dimensions and is defined to be:

$$R(t; \bar{\mathbf{x}}, \bar{\mathbf{d}}) = \{\mathbf{x} \in R^m : \bar{\mathbf{x}} - t\bar{\mathbf{d}} \leq \mathbf{x} \leq \bar{\mathbf{x}} + t\bar{\mathbf{d}}\}. \tag{1}$$

When the vectors $\bar{\mathbf{x}}$ and $\bar{\mathbf{d}}$ are clear, the rectangle in (1) is abbreviated $R(t)$. Observe from Figure 1 that as t increases from 0 to ∞, $R(t)$ expands from the point $R(0) = \{\bar{\mathbf{x}}\}$ to cover all of R^m. Thus, for any vector $\mathbf{y} \in R^m$, there is a real number $t \geq 0$ such that $\mathbf{y} \in R(t)$ and, in fact, there is a smallest such number, as stated in the following proposition:

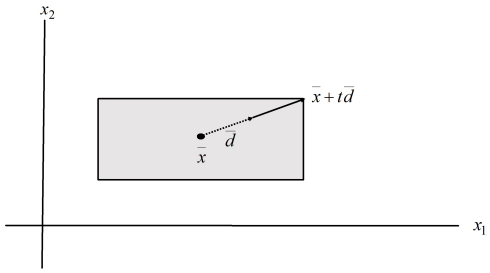

Figure 1: The rectangle of size t centered at $\bar{x}$ in the direction $\bar{d}$.

Proposition 2.1 For any vector $\mathbf{y} \in R^m$, the following real number $\bar{t} \geq 0$ has the property that $\mathbf{y} \in R(\bar{t})$ and, for all real numbers t with $0 \leq t < \bar{t}$, $\mathbf{y} \notin R(t)$:

$$\bar{t} = \max\left\{\frac{|\bar{x}_j - y_j|}{\bar{d}_j} : j = 1, \ldots, m\right\}.$$

The next proposition states that if $\bar{\mathbf{x}} \in R^m$ is strictly on one side of a hyperplane in R^m, then there is a largest rectangle centered at $\bar{\mathbf{x}}$ such that the entire rectangle is contained on the same side of the hyperplane as $\bar{\mathbf{x}}$.

Proposition 2.2 Given a nonzero vector $\mathbf{a} \in R^m$ and a scalar b, suppose that the point $\bar{x} \in R^m$ is in the half-space $H^> = \{\mathbf{x} \in R^m : \mathbf{a}^T\mathbf{x} > b\}$, that is, $\mathbf{a}^T\bar{\mathbf{x}} > b$. Then the following formula determines the largest value of $t > 0$ such that $R(t) \subset H^{\geq}$ (if $\mathbf{a}^T\mathbf{x} < b$, then the vector $-\mathbf{d}$ is used in place of $\mathbf{d}$ below):

$$\bar{t} = \frac{b - \mathbf{a}^\mathbf{T}\bar{\mathbf{x}}}{\mathbf{a}^\mathbf{T}\mathbf{d}}, \quad \text{where } d_j = \begin{cases} -\bar{d}_j & \text{if } a_j \geq 0 \\ +\bar{d}_j & \text{if } a_j < 0 \end{cases}$$

2.2 Preliminary Results for Quadratic Functions

Throughout this paper, Q is an $m \times m$ symmetric matrix, $\mathbf{q} \in R^m$, and p is a scalar. The matrix Q is *positive definite* (pd) if and only if for all vectors $\mathbf{x} \in R^m$ with $\mathbf{x} \neq \mathbf{0}$, $\mathbf{x}^T Q\mathbf{x} > 0$. In this case, Q is nonsingular and Q^{-1} is also pd. Moreover, the function $g : R^m \to R^1$ defined by $g(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T Q\mathbf{x} + \mathbf{x}^T\mathbf{q} - p$ is convex and $\nabla g(\mathbf{x})^T = \mathbf{x}^T Q + \mathbf{q}^T$.

Given vectors $\mathbf{x}, \mathbf{d} \in R^m$, the convex quadratic function on the line from $\mathbf{x}$ in the direction $\mathbf{d}$ is $h : R^1 \to R^1$ defined by $h(t) = \frac{1}{2}(\mathbf{x}+t\mathbf{d})^T Q(\mathbf{x}+t\mathbf{d}) + (\mathbf{x}+t\mathbf{d})^T\mathbf{q} - p$ and can be written in the form $at^2 + bt + c$, where

$$a = \frac{1}{2}\mathbf{d}^T Q\mathbf{d}. \quad b = \left(\mathbf{x}^T Q + \mathbf{q}^T\right)\mathbf{d}, \quad \text{and } c = \frac{1}{2}\mathbf{x}^T Q\mathbf{x} + \mathbf{x}^T\mathbf{q} - p.$$

2.3 Preliminary Results on Finding a Zero of a Convex Function

For a given convex function $g : R^m \to R^1$, a goal of this research is to develop an algorithm for allocating a simulation budget when the set Γ is defined as follows:

$$\Gamma = \{\mathbf{x} \in R^m : g(\mathbf{x}) \leq 0\}. \tag{2}$$

In so doing, it sometimes becomes necessary to solve a slightly modified version of the following zero-finding problem:

Problem 1: Given a convex function $h : R^1 \to R^1$, find a point $t^* \in R^1$ such that $h(t^*) = 0$.

Solow and Paparizzos (1989) developed a subgradient Newton method to solve Problem 1 that has some very desirable properties as a result of h being convex. In particular, starting anywhere, the algorithm either terminates finitely, in which case a zero of h is found or else there is no zero of h. Alternatively, if the algorithm generates an infinite sequence of points, then that sequence either converges to a zero of h or diverges to $\pm\infty$, in which case, h has no zero. Turning to the application here, for the convex function in (2), in the subsequent development it is sometimes necessary to solve the following problem:

Problem 2: Given a convex function $g : R^m \to R^1$, together with $\mathbf{x}, \mathbf{z} \in \mathbf{R^m}$ where $g(\mathbf{x}) > 0$ and $g(\mathbf{z}) < \mathbf{0}$ (or vice versa), find a point $\mathbf{y}$ on the line segment connecting $\mathbf{x}$ and $\mathbf{z}$ where $g(\mathbf{y}) = \mathbf{0}$.

Note that, by the continuity of g, the point $\mathbf{y}$ is known to exist. Furthermore, finding such a point is equivalent to finding a zero of the function $h : R^1 \to R^1$ defined by $h(t) = g(\mathbf{x} + t(\mathbf{z} - \mathbf{x}))$ in the interval [0, 1], thus giving rise to the following problem:

Problem 3: Given a convex function $h : R^1 \to R^1$ for which $h(0) > 0$ and $h(1) < 0$ (or vice versa), find a point $t^* \in [0, 1]$ such that $h(t^*) = 0$.

If t^* solves Problem 3, then $\mathbf{y} = \mathbf{x} + t^*(\mathbf{z} - \mathbf{x})$ satisfies $g(\mathbf{y}) = h(t^*) = 0$ and hence $\mathbf{y}$ solves Problem 2. Problem 3 is a modified version of Problem 1 in which a zero of h is sought in the interval [0, 1] which is known to contain such a point. In the event that $h(t)$ is a quadratic function, such as the $h(t)$ in Section 2.2, a value of t^* that solves Problem 3 is available from the quadratic formula. For a general convex function h, conveniently, Solow's algorithm works equally well on Problem 3 simply by starting the algorithm at the endpoint of the interval [0, 1] where the value of h is > 0.

2.4 Preliminary Results for Random Variables and Confidence Intervals

As mentioned in the Introduction, each system $i = 1, \ldots, r$ has associated with it m performance measures whose values are the unknown means of m random variables. For ease of exposition, all results presented

in this subsection apply to a single system; the superscript i is subsequently used to denote the appropriate quantity associated with system i. Let $\mathbf{X} = (X_1, \ldots, X_m)$ be the vector of m random variables associated with a given system and let $\mu = (\mu_1, \ldots, \mu_m)$ and $\sigma = (\sigma_1, \ldots, \sigma_m)$ be their means and standard deviations. Monte Carlo simulation is used to estimate the values of μ. That is, after n replications of the simulation, $\bar{\mathbf{x}} = (\bar{x}_1, \ldots, \bar{x}_m)$ are used as the current estimates of the means $\mu = (\mu_1, \ldots, \mu_m)$ for this system.

To simplify the exposition, for now, all results are presented assuming that the standard deviations $\sigma = (\sigma_1, \ldots, \sigma_m)$ are known and the m random variables are pairwise independent of each other; the case when the values of σ are unknown and the random variables could be dependent are discussed in Solow et al. (2020). With these simplifying assumptions, letting Φ be the cumulative distribution function of the standard normal random variable $Z \sim N(0,1)$, and given a value of α with $0 < \alpha < 1$, the $(1\text{-}\alpha) \times 100\%$ confidence interval for μ_j after completing n simulation replications is:

$$\bar{x}_j \pm t_\alpha \sigma_j, \quad \text{where } t_\alpha = z/\sqrt{n} \text{ and } z = \Phi^{-1}(1-\alpha/2). \tag{3}$$

Putting together the m independent confidence intervals in (3) results in the confidence rectangle of size t_α centered at $\bar{x}$ in the direction σ, namely,

$$R(t_\alpha; \bar{\mathbf{x}}, \sigma) = \{\mathbf{x} \in R^m : \bar{\mathbf{x}} - t_\alpha \sigma \leq \mathbf{x} \leq \bar{\mathbf{x}} + t_\alpha \sigma\}. \tag{4}$$

Observe that while it is correct to interpret (3) as being $(1\text{-}\alpha) \times 100\%$ confident that μ_j is in that confidence interval, the interpretation of the confidence rectangle is that you are $(1-\alpha)^m \times 100\%$ confident that the vector μ is in the confidence rectangle. In the ensuing algorithms, it is assumed that the analyst will provide the desired level of confidence, β. Thus, to be $(1\text{-}\beta) \times 100\%$ confident that the vector μ is in the confidence rectangle, it is necessary to use the value $\alpha = 1 - (1-\beta)^{1/m}$ in (3) and in (4). Furthermore, while the analyst could, in theory, provide a different confidence level for each system, it is assumed for simplicity that the same value is provided for all systems.

Having completed these preliminary results, we now present the intuition behind the classification algorithm with a budget of N simulation replications. The basic idea is for the analyst to choose a desired level of confidence for all systems, say β with $0 < \beta < 1$; to collect an initial number of samples (through independent simulation replications) from each system i; and then to construct a $(1\text{-}\beta) \times 100\%$ confidence rectangle centered at the sample mean $\bar{\mathbf{x}}^i$. As seen in Figure 2(a), if this confidence rectangle does not intersect the feasible region Γ, then system i is classified (perhaps incorrectly) as not in Γ; whereas, if this confidence rectangle is contained completely within Γ, then system i is classified (perhaps incorrectly) as in Γ. On the other hand, as seen in Figure 2(b), if this confidence rectangle intersects the feasible region Γ while the sample mean falls outside of Γ or if the confidence rectangle "spills out" of Γ while the sample mean falls inside Γ, the system cannot yet be classified with the desired level of confidence. In this case, the number of additional replications needed to shrink the confidence rectangle so that the shrunken rectangle lies completely outside, or completely inside, Γ is determined. These additional replications are then conducted until either every system is classified with $(1\text{-}\beta) \times 100\%$ confidence or until the budget is exhausted. The details of the algorithm are now developed for Γ defined by nonlinear inequalities.

3 A BUDGET ALLOCATION ALGORITHM WITH Γ DEFINED BY NONLINEAR INEQUALITIES

Throughout this section, it is assumed that Γ is defined by a finite collection of s nonlinear inequalities. Let $g_j : R^m \to R^1$ be convex functions for $j = 1, \ldots, s$ that define Γ as follows:

$$\Gamma = \{\mathbf{x} \in R^m : g_j(\mathbf{x}) \leq 0, j = 1, \ldots, s\}.$$

Suppose further that Γ has a nonempty interior Γ^o. In the subsequent analysis, the following convex function $g : R^m \to R^1$ is sometimes used:

$$g(\mathbf{x}) = \max_{j=1,\ldots,s} g_j(\mathbf{x}). \tag{5}$$

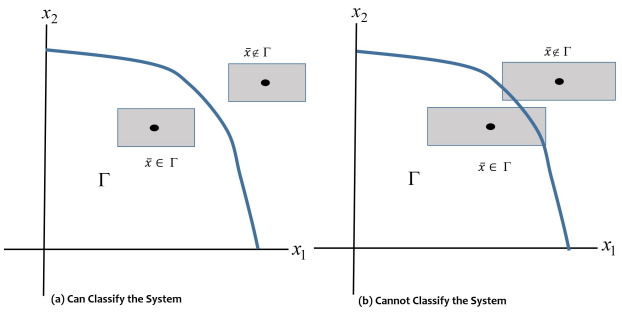

Figure 2: Confidence rectangles that allow and do not allow a system to be classified.

One special case of interest is when Γ is defined by a single convex quadratic inequality, in which case, $g(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T Q\mathbf{x} + \mathbf{x}^T\mathbf{q} - p$, for some positive definite symmetric matrix Q.

Throughout this section, the superscript i for the system is omitted. For a particular system, suppose that n_0 initial simulation replications have resulted in estimated means, $\bar{\mathbf{x}} \in R^m$, and that the following $(1\text{-}\beta) \times 100\%$ confidence rectangle has been constructed using the values of $\alpha = 1-(1-\beta)^{1/m}$, $z = \Phi^{-1}(1-\alpha/2)$, and $\tilde{t} = z/\sqrt{n_0}$:

$$R(\tilde{t}; \bar{\mathbf{x}}, \sigma) = \{\mathbf{x} \in R^m : \bar{\mathbf{x}} - \tilde{t}\sigma \leq \mathbf{x} \leq \bar{\mathbf{x}} + \tilde{t}\sigma\}.$$

The foregoing confidence rectangle is hereafter referred to more simply as $R(\tilde{t})$. How the algorithm proceeds depends on whether $\bar{\mathbf{x}} \in \Gamma$ or not, as is now addressed.

3.1 Analysis When $\bar{x} \notin \Gamma$

In this case, if $R(\tilde{t}) \cap \Gamma = \emptyset$, then the system is classified as not in Γ. Determining whether $R(\tilde{t}) \cap \Gamma = \emptyset$ or not can be accomplished by determining whether the following system is feasible:

$$\begin{aligned} &g_j(\mathbf{x}) \leq 0, j = 1, \ldots, s \\ &\mathbf{x} \in R(\tilde{t}) \end{aligned} \tag{6}$$

If (6) is not feasible, then $R(\tilde{t}) \cap \Gamma = \emptyset$; so, the system is classified as not in Γ with $(1\text{-}\beta) \times 100\%$ confidence. Solving (6) in general requires using a nonlinear programming package. However, if that computer package is capable of minimizing a nondifferentiable convex objective function, then another approach for determining whether (6) is feasible is to solve the following optimization problem where g is defined in (5):

$$\begin{aligned} \min \quad & g(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{x} \in R(\tilde{t}) \end{aligned} \tag{7}$$

As g is continuous and $R(\tilde{t})$ is nonempty and compact, (7) has an optimal solution, say $\tilde{\mathbf{x}} \in R^m$, from which it follows that $R(\tilde{t}) \cap \Gamma = \emptyset$ if and only if $g(\tilde{\mathbf{x}}) > 0$. It is also worth noting that, for the special case when Γ is defined by a single convex quadratic constraint, (7) is a convex quadratic programming problem that can be solved in a finite number of iterations using a variety of algorithms (see Solow and Li (2007)).

In summary, if (6) is infeasible, then $R(\tilde{t}) \cap \Gamma = \emptyset$ and the system is classified as not in Γ with $(1\text{-}\beta) \times 100\%$ confidence. On the other hand, if (6) is feasible, then $R(\tilde{t}) \cap \Gamma \neq \emptyset$; hence, the system cannot yet be classified with the desired level of confidence. In this case, the idea is to determine the number of additional simulation replications that would be needed so that the resulting shrunken rectangle, still centered at $\bar{\mathbf{x}}$, no longer intersects Γ. To that end, the values of $t \geq 0$ for which $R(t)$ has at least one point in Γ play a central role in the subsequent analysis, so let

$$T = \{t \geq 0 : R(t) \cap \Gamma \neq \emptyset\}.$$

Observe that the foregoing set $T \neq \emptyset$ because $\tilde{t} \in T$. Furthermore, by definition, T is bounded below by 0. As such, T has an infimum, say

$$t^* = \inf T = \inf\{t \geq 0 : R(t) \cap \Gamma \neq \emptyset\}.$$

In words, t^* implies that a rectangle of any size $t < t^*$ will not intersect Γ; hence, this system would be classified as not in Γ. This means that, had $\tilde{t}$ been less than t^*, the system would have been classified as not in Γ. Equivalently stated, had the number of simulations replications, n_0, been at least $(z/t^*)^2$, the system would have been classified as not in Γ. Thus, if this system is not classified, then the number of additional simulation replications to be performed on this system is:

$$\Delta n = \left\lceil \left(\frac{z}{t^*}\right)^2 - n_0 \right\rceil. \tag{8}$$

One possible approach for finding t^* is now presented.

An Optimization Approach for Finding t^*: One way to find t^* is to solve the following optimization problem:

$$\begin{array}{lrcl} \min & t & & \\ \text{s.t.} & \mathbf{x} & \in & \Gamma \cap R(t) \\ & t & \geq & 0 \end{array} \tag{9}$$

For example, when $\Gamma = \{\mathbf{x} \in R^m : A\mathbf{x} \leq b\}$, the optimization problem in (9) becomes a bounded and feasible linear program. For the nonlinear case when $\Gamma = \{\mathbf{x} \in R^m : g_j(\mathbf{x}) \leq 0, j = 1, \ldots, s\}$, one could, in theory, find t^* by using a nonlinear programming package to solve the following convex optimization problem:

$$\begin{array}{ll} \min & t \\ \text{s.t.} & g_j(\mathbf{x}) \leq 0 \\ & \bar{\mathbf{x}} - t\boldsymbol{\sigma} \leq \mathbf{x} \leq \bar{\mathbf{x}} - t\boldsymbol{\sigma} \\ & t \geq 0 \end{array} \tag{10}$$

Observe that a feasible solution $(\tilde{\mathbf{x}}, \tilde{t})$ to (6) is an initial feasible solution for (10) because $\tilde{\mathbf{x}} \in R(\tilde{t})$ and $g_j(\tilde{\mathbf{x}}) \leq 0, j = 1, \ldots, s$.

3.2 Analysis When $\bar{x} \in \Gamma$

When the current sample mean $\bar{\mathbf{x}} \in \Gamma$, it is first necessary to determine whether the confidence rectangle $R(\tilde{t})$ is contained entirely in Γ, for then the system can be classified as in Γ with $(1\text{-}\beta) \times 100\%$ confidence. The following proposition provides one method for checking whether $R(\tilde{t}) \subset \Gamma$:

Proposition 3.1 If Γ is convex, then for any $t > 0$, $R(t) \subset \Gamma$ if and only if every extreme point of $R(t)$ is in Γ.

As a result of Proposition 3.2, to check whether $R(t) \subset \Gamma$, one must check if all 2^m extreme points of $R(t)$ are in Γ. When Γ is defined by linear inequalities, an alternative approach for checking if $R(t) \subset \Gamma$ can be done using the c constraints that define Γ, one at a time, as described in Solow et al. (2020). Unfortunately, when Γ is defined by nonlinear inequalities, that alternative is not available; however, the following optimization approach can be used, where g is defined as in (5):

$$\begin{array}{ll} \max & g(\mathbf{x}) \\ \text{s.t.} & \mathbf{x} \in R(t) = R(t;\ \bar{\mathbf{x}},\ \mathbf{d}) = \{\mathbf{x} \in R^m : \bar{\mathbf{x}} - t\mathbf{d} \leq \mathbf{x} \leq \bar{\mathbf{x}} + t\mathbf{d}\}. \end{array} \tag{11}$$

The following proposition shows how to use an optimal solution to (11) to determine if $R(t) \subset \Gamma$:

Proposition 3.2 If $\Gamma = \{\mathbf{x} : g(\mathbf{x}) \leq 0\}$ and $\mathbf{x}^*$ is an optimal solution to (11), then $R(t) \subset \Gamma$ if and only if $g(\mathbf{x}^*) \leq 0$.

As a result of Proposition 3.2, it is necessary to solve (11). Unfortunately, even when g is a single, convex, quadratic function, (11) is NP-hard. However, there is one special case when it could be possible to obtain an optimal solution to (11) efficiently. In particular, if Γ consists of a single, *separable*, convex inequality, say $g(\mathbf{x}) = \sum_i g_i(x_i) \leq 0$, then (11) decomposes into the following collection of m one-dimensional subproblems, whose optimal solutions, say x_i^*, collectively form the optimal solution to (11):

$$\begin{array}{ll} \max & g_i(x_i) \\ \text{s.t.} & \bar{x}_i - td_i \leq x_i \leq \bar{x}_i + td_i \end{array}$$

In the event that the confidence rectangle $R(\tilde{t})$ is not contained entirely in Γ, it is then necessary to determine the size of a reduced rectangle so that the shrunken rectangle, still centered at $\bar{\mathbf{x}}$, would be entirely in Γ. To that end, consider the following set of real numbers:

$$T = \{t \geq 0 : R(t) \subset \Gamma\}.$$

Observe that $0 \in T$ because $R(0) = \{\bar{\mathbf{x}}\} \subset \Gamma$ and so $T \neq \emptyset$. Furthermore, T is bounded above because, for large enough t, $R(t)$ will contain a point that is not in Γ. As such, T has a supremum, say

$$t^* = \sup T = \sup\{t \geq 0 : R(t) \subset \Gamma\}. \tag{12}$$

In words, t^* implies that a rectangle of any size $t > t^*$ will not be contained entirely in Γ. Once the value of t^* is found, the number of additional simulation repetitions needed to shrink the rectangle so that the shrunken rectangle would be entirely in Γ is given in (8). However, to ensure that $t^* > 0$, it is assumed that $\bar{\mathbf{x}} \notin \partial\Gamma$, that is, that $g(\bar{\mathbf{x}}) < 0$.

When $\Gamma = \{\mathbf{x} \in R^m : g_j(\mathbf{x}) \leq 0, j = 1, \ldots, s\}$, it is possible to determine t^* in (12) by computing a number, t^k, associated with each extreme point $\hat{\mathbf{x}}^\mathbf{k}$ of $R(\tilde{t})$ that is not in Γ, of which there is at least one (otherwise, the system would have been classified as in Γ). The value of t^k represents the size of the rectangle that would pull extreme point $\hat{\mathbf{x}}^\mathbf{k}$ of $R(\tilde{t})$ back toward $\bar{x}$ so that the resulting extreme point is in Γ. To determine t^k using $g(\mathbf{x})$ as defined in (5), note that each extreme point $\hat{\mathbf{x}}^\mathbf{k}$ of $R(\tilde{t})$ that is not in Γ satisfies $g(\hat{\mathbf{x}}^\mathbf{k}) > 0$. As $\bar{\mathbf{x}} \in \Gamma^o$, $g(\bar{\mathbf{x}}) < 0$. Consequently, by the continuity of g, there is a point $\hat{\mathbf{x}}$ on the line segment from $\bar{\mathbf{x}}$ to $\hat{\mathbf{x}}^\mathbf{k}$ such that $g(\hat{\mathbf{x}}^\mathbf{k}) = \mathbf{0}$. As before, finding $\hat{\mathbf{x}}^\mathbf{k}$ is equivalent to finding a value of $t^k \in [0,1]$ so that the function $h : R^1 \to R^1$ defined by

$$h(t) = g(\bar{\mathbf{x}} + t(\hat{\mathbf{x}}^\mathbf{k} - \bar{\mathbf{x}}))$$

satisfies $h(t^k) = 0$ for then the point on the line segment from $\bar{x}$ to $\hat{\mathbf{x}}^\mathbf{k}$ where the value of g is zero is $\bar{x} + t^k(\hat{\mathbf{x}}^\mathbf{k} - \bar{x})$. In other words, it is necessary to find a zero of the real-valued convex function h on the interval [0, 1]. How this can be done is discussed in Section 2.3 for the special case when Γ is defined by a single convex quadratic constraint.

Having found the value for t^k associated with each extreme point $\hat{\mathbf{x}}^\mathbf{k}$ of $R(\tilde{t})$ that is not in Γ, the desired value of $t^* = \sup T$ is given in the following proposition.

Proposition 3.3 Suppose $\bar{\mathbf{x}} \in \Gamma$, but the confidence rectangle $R(\tilde{t})$ is not contained in Γ. If, for every extreme point $\hat{\mathbf{x}}^\mathbf{k} \in R(\tilde{t})$ that is not in Γ, $t^k \in [0,1]$ satisfies $g(\bar{\mathbf{x}} + t^k(\hat{\mathbf{x}}^\mathbf{k} - \bar{\mathbf{x}})) = 0$, then

$$t^* = \min\{t^k : \hat{\mathbf{x}}^\mathbf{k} \text{ is an extreme point of } R(\tilde{t}) \text{ that is not in } \Gamma\} = \textit{sup } T.$$

3.3 Recap

The following is a summary of the heuristic algorithm for allocating a budget of N simulation replications so as to classify each of the r systems as in or not in Γ when $\Gamma = \{\mathbf{x} \in R^m : g(\mathbf{x}) \leq 0\}$, $\Gamma^o \neq \emptyset$, and each sample mean $\bar{\mathbf{x}}^i \notin \partial\Gamma$:

A Finite Budget Algorithm When Γ is Defined by Nonlinear Inequalities

Step 0. Obtain the desired confidence level β with $0 < \beta < 1$ from the analyst and compute $\alpha = 1-(1-\beta)^{1/m}$. For each system i, perform some number, say n_0^i, of initial replications of the simulation and compute the sample mean $\bar{\mathbf{x}}^i$ together with the following value of t^i that determines the size of the associated confidence rectangle:

$$t^i = z/\sqrt{n_0^i}, \quad \text{where } z = \Phi^{-1}(1-\alpha/2).$$

Step 1(a). For each system i for which $\bar{\mathbf{x}}^i \notin \Gamma$, determine whether the inequalities in (6) are feasible; if not, then $R(t^i) \cap \Gamma = \emptyset$ and so system i is classified as not in Γ with $(1\text{-}\beta) \times 100\%$ confidence. Otherwise, solve the optimization problem in (10). Then use (8) to compute the number of additional replications, Δn^i, needed to shrink the confidence rectangle so that this system would have been classified as not in Γ.

Step 1(b). For each system i for which $\bar{\mathbf{x}}^i \in \Gamma$, if all 2^m extreme points of $R(t^i)$ are in Γ, then system i is classified as in Γ with $(1\text{-}\beta) \times 100\%$ confidence. Otherwise, for each extreme point $\hat{\mathbf{x}}^k \notin \Gamma$, compute t^k so that $\hat{\mathbf{x}} = \bar{\mathbf{x}}^i + t^k(\hat{\mathbf{x}}^{\mathbf{k}} - \bar{\mathbf{x}}^i)$ satisfies $g(\hat{\mathbf{x}}) = 0$ and then let t^* be the minimum of all such numbers. Use (8) to compute the number of additional replications, Δn^i, needed to shrink the confidence rectangle so that this system would have been classified as in Γ.

Step 2. If all systems are classified, stop. Otherwise, sort the Δn^i in increasing order, for all the unclassified systems i. Starting with the smallest one, if Δn^i exceeds the remaining budget, then go to Step 3 to classify every remaining unclassified system; otherwise, perform Δn^i additional simulation replications for system i, compute the new mean $\bar{\mathbf{x}}^i$ and t^i for the confidence rectangle, and reduce the budget accordingly. Now return to Step 1.

Step 3. For each remaining unclassified system i, if $\bar{\mathbf{x}}^i \in \Gamma$, then classify system i as in Γ while if $\bar{\mathbf{x}}^i \notin \Gamma$ then classify system i as not in Γ, with confidence less than $(1\text{-}\beta) \times 100\%$.

4 COMPUTATIONAL EXPERIMENTS

To assess the effectiveness of the algorithms developed in Section 3, preliminary computational experiments are performed in VBA to estimate the following four *performance measures*, whose averages are taken over a collection of 50 problems, each containing $r = 10$ systems and $m = 4$ random variables:

%CC w β	=	the average number of systems correctly classified with confidence $(1-\beta) \times 100\%$.
%CC w/o β	=	the average number of systems correctly classified with confidence less than $(1-\beta) \times 100\%$.
%IC w β	=	the average number of systems incorrectly classified with confidence $(1-\beta) \times 100\%$.
%IC w/o β	=	the average number of systems incorrectly classified with confidence less than $(1-\beta) \times 100\%$.

In addition to the specific way in which the data are generated, the values of the foregoing performance measures depend on the adjustable parameters, namely the confidence level, $1-\beta$, the computational budget, N, and the number of initial simulation replications, n_0, performed for each system.

For each of the 80%, 90%, 95%, and 99% confidence levels, data are generated for a specific problem and a *target budget* is computed, based on the data. Each such problem is then solved four times, using 50%, 100%, 150%, and 200% of the target budget as the actual budget, N. To examine the effect of the number of initial simulation replications on the performance measures, each such budget is then divided equally among the 10 systems and then 10%, 30%, 50%, and 75% of the budget is used for the initial

number of replications. Computational experiments that were performed for the setting where Γ is defined by a sphere are discussed next.

4.1 Γ is Defined by a Sphere

In this section, Γ is defined as the following sphere:

$$\Gamma = \{\mathbf{x} \in R^m : ||\mathbf{x}||^2 \leq r^2\}.$$

Each of the 50 problems contained 10 systems consisting of $m = 4$ random variables $X_j \sim U[a_j, b_j]$, where a_j and b_j are as yet undetermined. The radius of the sphere is set arbitrarily to $r = 16$. The means of half of the 10 systems are generated in the sphere and half are outside. To that end, a random point $\mathbf{y}$ on the surface of the sphere in the positive orthant is first created, together with each standard deviation $\sigma_j \sim U[0.5, 1.0]$. Using a random value of $n_j \sim U[100, 200]$, the mean μ_j of the random variable X_j is then computed as

$$\mu_j = y_j \pm z * \sigma_j / \sqrt{n_j},$$

depending on whether the mean should be in Γ or not and the target budget is set to $\sum_j \lfloor n_j/3 \rfloor$. Now that the values of μ_j and σ_j are known, the value of the half-width e_j of the interval $[a_j, b_j]$ is computed from

$$\sigma_j^2 = \frac{(b_j - a_j)^2}{12} = \frac{\left(\frac{b_j - a_j}{2}\right)^2}{3} = \frac{e_j^2}{3} \quad \text{so} \quad e_j = \sqrt{3}\sigma_j$$

and then

$$a_j = \mu_j - e_j \quad \text{and} \quad b_j = \mu_j + e_j.$$

In almost all of the computational experiments performed with these specifications, using 30%, 50%, and 75% of the target budget for the initial simulations replications was better on average than using 10%. This is seen in Figure 3(a) for the 95% confidence level with similar results for other levels of confidence. On the other hand, Figure 3(b) shows that, if the objective is to correctly classify as many systems as possible, regardless of the level of confidence, then it is best to use 75% of the target budget for performing initial simulation replications. For the subsequent discussion here, computational results are presented only for the initial number of simulation replications being 30% of the target budget.

Turning to the effect of the size of the budget on the performance measures, each of the four bars in Figure 4 shows the average values of the correctly classified systems using $\beta = 0.05$ when the actual budget is 50%, 100%, 150%, and 200%, respectively, of the target budget. There it is seen that %CC w β increases from a value of about 55% when the actual budget is 50% of the target budget to about 87% when the actual budget is 200% of the target budget. Likewise, %CC w/o β decreases from a value of about 37% when the actual budget is 50% of the target budget to about 13% when the actual budget is 200% of the target budget. As the fractions of incorrectly classified systems, both with and without the specified level of confidence, are extremely close to 0, those values are not reported.

5 CONCLUSIONS

We provide a heuristic for addressing the feasibility problem in which the feasible region, Γ, is defined by a finite collection of nonlinear inequalities. This approach allows the user to choose a desired level of confidence with which a system is correctly classified as in Γ or not in Γ. The algorithm then uses appropriate-sized confidence rectangles centered at the estimated means to decide when a system can be classified as in Γ or not in Γ, with the specified level of confidence. While the worst-case behavior could potentially be bad, computational experiments show that the performance of the algorithm on randomly generated problems is satisfactory.

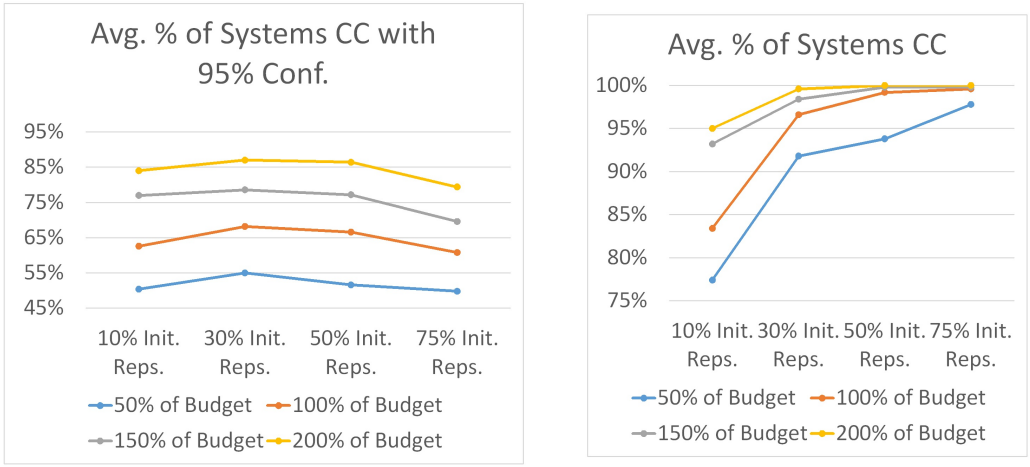

Figure 3: Average number of correctly classified systems as a function of the initial replications for a uniform distribution with $\beta = 0.05$.

There are several natural extensions of this work that are currently under study. The first extension is the setting where variances are not known. In this case, it is possible to estimate the unknown variances from the initial sample of size n_0. The challenge, however, is to ensure the invertibility of the estimated covariance matrix. The second extension is the setting where we relax the assumption that the random variables are independent. In this setting, it is possible to use a translation of the coordinate axes as described in Solow et al. (2020).

REFERENCES

Andradottir, S., and S. H. Kim. 2010. "Fully sequential procedures for comparing constrained systems via simulation". *Naval Research Logistics* 57:403–421.

Batur, D., and S. H. Kim. 2010. "Finding feasible systems in the presence of constraints on multiple performance measures". *ACM Transactions on Modeling and Computer Simulation* 20:118–130.

Gao, S., and W. Chen. 2017. "Efficient feasibility determination with multiple performance measurement constraints". *IEEE Transactions on Automatic Control* 62:113–122.

He, J., and S.-H. Kim. 2019. "A new reward function for Bayesian feasibility determination". In *Proceedings of the 2019 Winter Simulation Conference*, edited by N. Mustafee, K.-H. G. Bae, S. Lazarova-Molnar, M. Rabe, C. Szabo, P. Haas, and Y.-J. Son, 3480–3491. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Hunter, S., and B. McClosky. 2016. "Maximizing quantitative traits in the mating design problem via simulation-based Pareto estimation". *IISE Transactions* 48:565–578.

Lee, L. H., N. A. Pujowidianto, C. H. Chen, and C. M. Yap. 2012. "Asymptotic simulation budget allocation for selecting the best design in the presence of stochastic constraints". *IEEE Transactions on Automatic Control* 57:2940–2945.

Pasupathy, R., S. R. Hunter, N. A. Pujowidianto, L. H. Lee, and C. H. Chen. 2014. "Stochastically constrained ranking and selection via SCORE". *ACM Transactions on Modeling and Computer Simulation* 25:1 – 26.

Santner, T. J., and A. C. Tamhane. 1984. "Designing experiments for selecting a normal population with a large mean and a small variance". In *Design of Experiments - Ranking and Selection: Essays in Honor of Robert E. Bechhoffer*, edited by T. J. Santner and A. C. Tamhane, 179–198. New York: Marcel-Dekker.

Solow, D., and H. Li. 2007. "A constrained optimization approach to solving certain systems of convex equations". *European Journal of Operational Research* 176:1334 – 1347.

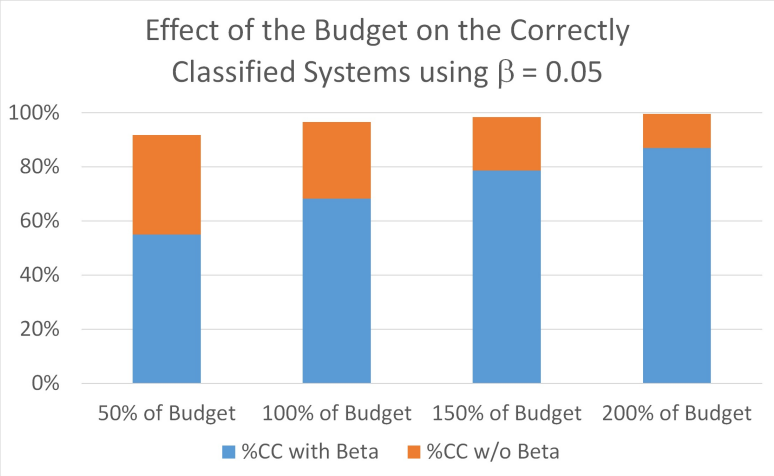

Figure 4: Effect of the budget on correctly classified systems for the uniform distribution for the 95% confidence level.

Solow, D., and K. Paparizzos. 1989. "A finite improvement algorithm for the linear complementarity problem". *European Journal of Operational Research* 3:305 – 324.

Solow, D., R. Szechtman, and E. Yücesan. 2020. "Novel approaches to feasibility determination". *ACM Transactions on Modeling and Computer Simulation*:Forthcoming.

Szechtman, R., and E. Yücesan. 2008. "A New perspective on feasibility determination". In *Proceedings of the Winter Simulation Conference*, edited by R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, 192–199. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Szechtman, R., and E. Yücesan. 2016. "A Bayesian approach to feasibility determination". In *Proceedings of the Winter Simulation Conference*, edited by T. Roeder, P. Frazier, R. Szechtman, E. Z. T. Huscka, and S. Chick, 782–790. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Xie, J., and P. I. Frazier. 2013. "Sequential Bayes-optimal policies for multiple comparisons with a known standard". *Operations Research* 61:1174–1189.

AUTHOR BIOGRAPHIES

DANIEL SOLOW is a professor in the Department of Operations of the Weatherhead School of Management at Case Western Reserve University in Cleveland, Ohio. He holds a PhD in Operations Research from Stanford University. His research areas include deterministic optimization, linear and nonlinear programming, combinatorial optimization and modeling and analysis of complex systems. His email address is dxs8@case.edu.

ROBERTO SZECHTMAN received his Ph.D. from Stanford University, and currently is a faculty member in the Operations Research Department at the Naval Postgraduate School. His research interests include applied probability and military operations research. His e-mail address is rszechtm@nps.edu.

ENVER YÜCESAN is the Abu Dhabi Commercial Bank Chair in International Management and a professor in the Technology and Operations Management Area at INSEAD. He holds a Ph.D. in Operations Research from Cornell University. Using simulation, he addresses problems in agriculture, supply chain management, and social media. His email address is enver.yucesan@insead.edu.